1 Running SC302Bug Version 3.1

SC302Bug is a software package that communicates with the SC302 Application Development System board (SC302ADS). The software uses word, I/O mode accesses.

All numbers are assumed to be in hexadecimal notation.

Copy sc302bug.exe to the desired directory location. SC302Bug is then run from a DOS command line:

```
c:\[path]>sc302bug
              SC302Bug Version 3.1
        This software uses word I/O mode accesses only.
        It also assumes HEX notation!

        You must enter the PnP I/O Base Address and EEPROM type.
        Enter ---> s <value> <s|x|z>
        Example:  SC302Bug> s 2000 x
SC302Bug>
```

2 Commands

Entering ***help*** or ***?*** at the SC302Bug prompt provides the following list of supported commands:

```
SC302Bug> ?
q                Quits the debugger
play             Play a script file
pause            Allows 'pausing' in script file playback
*                Script file comment
help             prints this display
?                prints this display
peekio           read a value from I/O space
pokeio           write a value to I/O space
peekscp          read a value from the SCP Port
pokescp          write a value to the SCP Port
dumpt            display all T interface registers
dumpu            display all U interface registers
selftest                 execute the board self test routines
s                set I/O Base and EEPROM type of installed card
o                write a value to SC302 I/O space
i                read a value from SC302 I/O space
peeke            read a value from the EEPROM
pokee            write a value to the EEPROM
setpnp           set the PNP registers of the installed card
```

2.1 s Command

The ***s*** command is used to set the SC302ADS card’s I/O Base Address given the card by the Plug and Play manager.

Before using most commands, the I/O Base Address must be set:

```
SC302Bug>help s
```

```
S <ADDRESS> <S|X|Z> - set I/O Base and EEPROM type of installed card
```

When the I/O Base Address is entered, the software will “test” this location by writing to a SC302 memory register and then reading back from it to see if the write was successful.

For example, if your I/O Range is 2000-2003, the I/O Base Address is 2000, and:

```
SC302Bug>s 1900
This address does not appear to be correct. Try s command again...
SC302Bug>s 2000
This I/O Base Address appears to work....
SC302Bug>
```

The EEPROM type can be a Xicor X25020 (tested), a Xicor X25080 (tested), or an SGS-Thomson ST93C46A (not tested). Enter X for the Xicor X25020, Z for the X25080, or S for the SGS-Thomson part. The Xicor and SGS-Thomson devices have opposite chip select signal levels.

2.2 o Command

The ***o*** command is used to write consecutive data words to SC302 Memory Space.

```
SC302Bug>? o
O <OFFSET> <DAT1>...<DAT20> - write the DATA WORDS to SC302 I/O space
```

OFFSET is the address offset from the I/O Base Address set using the **s** command.

For example, to “point” the ADPTR to the Port A Control Register (PACNT) in the SC302 Memory Space, enter:

```
SC302Bug>o 81e
Writing just to ADPTR...
        outputw(gioBase=2000; adrptr=081e)
```

Another example is to write 4 consecutive data words to initialize Transmit Buffer Descriptor (TxBD) data at DPR location 200:

```
SC302Bug>o 200 0102 0304 0506 0708
        outputw(gioBase=2000; adrptr=0200)
              outputw(gioBase+0x02=2002; dport=0102)
              outputw(gioBase+0x02=2002; dport=0304)
              outputw(gioBase+0x02=2002; dport=0506)
              outputw(gioBase+0x02=2002; dport=0708)
```

2.3 i Command

The ***i*** command is used to read data words from SC302 Memory Space.

```
SC302Bug>help i
I <OFFSET> <num. words> - read word(s) from SC302 I/O space
```

The default number of words to read is 1.

An example of reading the TxBD written above is:

```
SC302Bug>i 0200 4
        outputw(gioBase=2000; adrptr=0200)
                    inpw(gioBase+0x02=2002; dport=0102)
                    inpw(gioBase+0x02=2002; dport=0304)
                    inpw(gioBase+0x02=2002; dport=0506)
                    inpw(gioBase+0x02=2002; dport=0708)
```

Another example using the **i** command is reading the current status of ADPTR and DPORT by executing an **i** at the I/O Base Address:

```
SC302Bug>o 08b0 1500
        outputw(gioBase=2000; adrptr=08b0)
              outputw(gioBase+0x02=2002; dport=1500)
SC302Bug>i 2000
       inp(gioBase+0x00=2000;          adrptr=b0)
       inp(gioBase+0x01=2001; adrptr=08)
       inp(gioBase+0x02=2002;          dport=00)
       inp(gioBase+0x03=2003;  dport=15)
```

2.4 peekio, pokeio Commands

These commands can be used to read/write data bytes or words from/to PC I/O space. They do not reference the SC302 I/O Base Address.

```
SC302Bug>? peekio
PEEKIO [B|W] <LOC> - read a value from I/O space

SC302Bug>help pokeio
POKEIO [B|W] <LOC> <VAL> - write a value to I/O space
```

2.5 peekscp, pokescp Commands

These command read/write data from the U- or S/T-Chip using the SC302 Serial Communication Port (SCP) pins, as defined on the SC302ADS board.

```
SC302Bug>help peekscp
PEEKSCP <U|T> <N|B|O|R> <$LOC> - read a value from the SCP Port

SC302Bug>help pokescp
POKESCP <U|T> <N|B|O|R> <LOC> <VAL> - write a value to the SCP Port
```

These commands are a shortcut for accessing the SCP registers of the U- or S/T-Chip. The second parameter is the register type that you are accessing:

- B - Byte
- N - Nibble
- O - Overlay
- R - Register (only R6 on the S/T interface)

To read the revision number of the U-Chip on the SC302ADS (the revision number is found in byte register 15), enter:

```
SC302Bug>peekscp u b f
Byte Register 0f = 46
```

2.6 peeke, pokee Commands

These commands are used to read/write to/from a EEPROM device connected to the SC302.

```
SC302Bug>help peeke
PEEKE <r> [$LOC] - read a value from the EEPROM
SC302Bug>help pokee
POKEE <$LOC> <$Value>- write a value to EEPROM
```

2.6.1 peeke command

The peeke command takes on many flavors: ***peeke r***, ***peeke***, and ***peeke <$LOC>***.

The ***peeke r*** command allows the user to display the Xicor EEPROM Status Register:

```
SC302Bug> peeke r
X25020 Status Register = 00
```

Entering just ***peeke*** will perform a EEPROM dump of the contents of EEPROM memory.

```
SC302Bug> peeke
Byte 00 = 00
Byte 01 = 03
.
.
.
```

To see just one byte of EEPROM, enter ***peeke <$LOC>*** where $LOC is the address to be displayed.

```
SC302Bug>peeke 1
byte register 01 = 03
```

2.6.2 pokee command

The ***pokee*** command also has some derivatives: ***pokee <$Value>***, and ***pokee <$LOC> <$Value>***.

Note

The ***pokee*** command by itself has been “absorbed” into the other ***pokee*** commands.

Any ***pokee*** command will program the Xicor Status Register, if x or z is chosen with the ***s*** command, so that no regions of the EEPROM address array are write protected (BP1 and BP0 cleared). This may be required when using a blank Xicor part, as the EEPROM array may be received write protected.

Using ***pokee <$LOC> <$Value>*** will program the EEPROM Address with the entered value, then read and display it back.

```
SC302Bug>pokee 50 aa
X25020 Status Register = ff
X25020 Status Register = ff
X25020 Status Register = 00
              byte register 50 = aa
```

While the actual ***pokee*** write is being performed, the status update of the EEPROM will be displayed (the Status Register for the Xicor part, or the Busy/Ready indication of the 93C46.

NOTE

> The 93C46C part can not be programmed by SC302Bug. This EEPROM requires exactly 18 clocks when programming a byte, and the SCP only works in 8 clock increments.

Using ***pokee <$Value>*** will continue programming data from the previous programmed address plus one, i.e., if the last programmed address is 50, the next programmed address will be 51. This allows sequential programming of data without having to re-enter the address.

```
SC302Bug>pokee bb
EEPROM Status Register = ff
EEPROM Status Register = ff
EEPROM Status Register = ff
EEPROM Status Register = ff
EEPROM Status Register = 00
              byte register 51 = bb
```

2.7 dumpu,dumpt Commands

These command will display all of the registers of the U- or S/T-Chip. This includes the nibble and overlay registers.

```
SC302Bug>help dumpu
DUMPU - display all U interface registers
SC302Bug>help dumpt
DUMPU - display all T interface registers
```

When entering the **dumpu** command, one can also see the U-Chip revision number when the byte registers are being displayed.

2.8 selftest Command

The ***selftest*** command tests the SC302ADS board by performing the following tests:

—INTLOOP - internal SC302 IDL loopback on SCC1
—UALIVE - Reads the U-chip version number
—ULOOP - Performs U-interface IDL loopback
—TALIVE - Writes and reads back data from Byte Register 5

The selftest uses files stored in a selftest directory found in the SC302Bug working directory (i.e., c:\[path]\sc302bug\selftest). The files are iidlloop.txt and uidlloop.txt. Copies of these files with more **pause** commands in place for script use are found in the scripts directory (i.e., c:\[path]\sc302bug\scripts).

```
SC302Bug>help selftest
selftest - execute the board self test routines
```

2.9 help,? Commands

These commands show the command list or detailed information about the specified command.

```
SC302Bug>help?
HELP [CMD] - prints detailed help for command or summary list
```

These commands without the [CMD] argument display the list of current SC302Bug commands.

2.10 q Command

This command quits SC302Bug.

```
SC302Bug> ? q
Q - Quits the debugger
```

2.11 Scripting commands

The ***play***, ***pause***, and ***** commands are utilized using the scripting capabilities of SC302Bug.

To create a script file for SC302Bug to play, an ASCII text file, like the examples included on the distribution disk in the *scripts* directory or the *selftest* directory, must be generated.

NOTE

The last line in the script file needs to be a blank line with no characters.

2.11.1 play Command

The ***play*** command plays the identified script file. This file can be located using standard DOS pathnames. When playing a script, the SC302Bug> prompt changes to Playmode>.

```
SC302Bug>? play
PLAY <filename> - Play a script file
```

The text file is composed of valid SC302Bug commands and comments.

2.11.2 pause Command

The ***pause*** command stops executing the script file and waits until the RETURN key is pressed. If a “b” or “B” Is entered, playback of the script is aborted. Only script file playback can be aborted: SC302Bug commands (like **selftest**) can not be aborted.

```
SC302Bug>help pause
PAUSE - <Return> to Continue or <b> to Break
```

When playing a script file and a **pause** command is reached, the screen shows:

```
PlayMode>pause
PlayMode> <Return> to Continue or <b> to Break:
```

To cease running the script, enter “b” without the quotes.

2.11.3 * Command

The ***** is used to place comments in the script file. The * must be followed by a “blank” character.

```
SC302Bug>help *
* - Comment. Any line starting with a * and a space will be ignored
```

3 Programming a Blank EEPROM

When a blank or non-valid EEPROM device is located on the SCP lines at power up by the SC302, neither Win95 nor any other Plug and Play BIOS can configure the card with its desired ISA card configuration information: IRQ format, DMA resources, and I/O or Memory mapped address space.

Additionally, while all other Plug and Play devices have gone through the Plug and Play device configuration sequence, the SC302 will remain in the Configuration State

The “valid EEPROM data” that the SC302 is looking for is defined in Table 5-4. Specifying SE^{2}PROM, in the SC302 User’s Manual.

SC302Bug can be used to program this EEPROM based on the following constraints:

- All SCP devices attached to the SC302 reside only on the SCPENx pins.
- The SCPENx pins are pulled to their inactive state, so that making them input pins will not activate any “systems.”
- The EEPROM being programmed is either the Xicor X25020, Xicor X25080, or SGS-Thomson 93C46A.
- A free I/O range is known that will not conflict with any other device in the PC when writes are performed to this range.

The steps used to program these EEPROMs are as follows:

1. On power up, all Plug and Play cards are configured and have gone to the Wait for

Key State. The SC302 remains in Config State, meaning all Plug and Play Configuration Registers are available for programming.

2. Start SC302Bug.

```
c:\[path]> sc302bug
```

3. Run the ***setpnp*** command (described below), to program the SC302's Plug and Play Configuration Registers, providing an I/O Base Address in which to run SC302Bug commands. The ***setpnp*** command performs I/O writes to the Plug and Play configuration ports, the ADDRESS port at 0x0279, and the WRITE_DATA port at 0x0A79.

```
SC302Bug>setpnp 3 2000
```

4. Run the ***s*** command, setting the I/O Base Address used by SC302Bug for SC302 commands, and the EEPROM type.

```
SC302Bug>s 2000 x
```

5. ***Play*** the script file that holds the EEPROM data file. This file consists of ***pokee*** commands. Example script files are enclosed: c:\[path]\eeprom\iosrec.txt and c:\[path]\eeprom\memsrec.txt.

```
SC302Bug>play eeprom\iosrec.txt
SC302Bug>play eeprom\memsrec.txt
```

6. Alternatively, steps 2, 3, 4 and 5 can be in one script file. An example script file for this is also enclosed: c:\[path]\eeprom\iocomp.txt.

```
SC302Bug>play eeprom\iocomp.txt
```

7. Or finally, from the DOS command line, run sprog.exe for the 93C46A, xprog.exe for the X25020, or zprog.exe for the X25080. These versions of SC302Bug **only** program blank EEPROMS.The EEPROM script files they look for are: ssrec.txt for sprog.exe, xsrec.txt for xprog.exe, or zsrec.text for zprog.exe. The script file must be located in the EEPROM directory under the working directory. Copies of the EEPROM script files are found in the eeprom directory, c:\[path]\eeprom\.

   For example,

```
c:\[path]>xprog.exe
```

   looks for

```
c:\[path]\eeprom\xsrec.tst
```

3.1 setpnp command

The ***setpnp*** command is only used to establish an I/O location to place the SC302 card into, when a blank EEPROM is encountered on boot. Running this command on a SC302 that has been configured into the system does nothing.

```
SC302Bug>help setpnp
SETPNP <IRQ> <IOAddress> - set PNP registers
```

NOTE

The selected IRQ and I/O Address must be known to be conflict free in the PC the card is located in.

After running the ***setpnp*** command, one can run the ***s*** command, and then perform SC302Bug commands (like ***i*** or ***o***). This is useful when programming the EEPROM using the ***pokee*** command.

```
SC302Bug> setpnp 3 2000
IRQ is 03
IOAddress is 2000
Writing ISI...
Writing I/O Base Address...
Writing IRQ...
Activate!!!
Resetting the card...
```

4 Revision History

Version	Release Date	Comments
0.1	Dec 2, 1996	Preliminary Release for revision 1.0
0.2	Jan 30, 1997	For revision 2.0
0.3	April 21, 197	For revision 3.0
0.4	May 27, 1997	For revision 3.1